**Amendments to the Claims:**

This listing of claims will replace all prior versions and listings of claims in the application:

**Listing of Claims:**

1. (CURRENTLY AMENDED) A method for providing communication protocol layer independent security for data transmitted between a first process, executing on a first network node, and a second process, executing on a second network node, wherein the first network node and the second network node each support at least one common communication protocol layer, the method comprising the steps of:

establishing a Java secure ~~communication~~ channel between the first network node and the second network node;

establishing a first Java stream between the first process and the Java secure ~~communication~~ channel;

establishing a second Java stream between the second process and the Java secure ~~communication~~ channel;

in response to the data being written to the first Java stream, encrypting the data to generate encrypted data, the encrypting of the data being performed independent of any communication protocol layers used to transport the encrypted data from the first network node to the second network node;

causing the encrypted data to be transmitted from the first network node to the second network node according to the at least one communication protocol layer supported by the first and second network nodes; and

in response to the encrypted data being read from the second Java stream, decrypting the encrypted data to recover decrypted data which is identical to the data on the first network node before the data was written to the first Java stream, the decrypting of the encrypted data being performed independent of any communication protocol layers used to transport the encrypted data from the first network node to the second network node.

2. (PREVIOUSLY PRESENTED) The method of Claim 1, further including the steps of

performing a communication protocol layer specific encryption of data to be sent across the communication channel at the first network node, and

performing a communication protocol layer specific decryption of data received from the communication channel at the second network node.

3. (CANCELED)

4. (CURRENTLY AMENDED) The method of Claim 1, ~~wherein the communication channel is a Java secure channel, wherein the first stream is a Java stream,~~

~~wherein the second stream is a Java stream,~~

wherein the method further comprises the step of connecting the Java secure channel to a third Java stream, and

wherein the third Java stream provides for the transmission of data according to a specific communication protocol layer.

5. (CURRENTLY AMENDED) A non-transitory computer-readable medium carrying one or more sequences of one or more instructions for providing communication protocol layer independent security for data transmitted between a first process, executing on a first network node, and a second process, executing on a second network node, wherein the first network node and the second network node each support at least one common communication protocol layer, the one or more sequences of one or more instructions including instructions which, when executed by one or more processors, cause the one or more processors to perform the steps of:

establishing a Java secure ~~communication~~ channel between the first network node and the second network node;

establishing a first Java stream between the first process and the Java secure ~~communication~~ channel;

establishing a second Java stream between the second process and the Java

secure ~~communication~~ channel;

in response to the data being written to the first Java stream, encrypting the data to generate encrypted data, the encrypting of the data being performed independent of any communication protocol layers used to transport the encrypted data from the first network node to the second network node;

causing the encrypted data to be transmitted from the first network node to the second network node according to the at least one communication protocol layer supported by the first and second network nodes; and

in response to the encrypted data being read from the second Java stream, decrypting the encrypted data to recover decrypted data which is identical to the data on the first network node before the data was written to the first stream, the decrypting of the encrypted data being performed independent of any communication protocol layers used to transport the encrypted data from the first network node to the second network node.

6. (CURRENTLY AMENDED) The non-transitory computer-readable medium of Claim 5, wherein the non-transitory computer-readable medium further includes instructions for performing the steps of

performing a communication protocol layer specific encryption of the data on the first network node, and

performing a communication protocol layer specific decryption of the data on the second network node.

7. (CANCELED)

8. (CURRENTLY AMENDED) The non-transitory computer-readable medium of Claim 5, ~~wherein the communication channel is a Java secure channel,~~

~~wherein the first stream is a Java stream,~~

~~wherein the second stream is a Java stream,~~

wherein the non-transitory computer-readable medium further includes instructions for connecting the Java secure channel to a third Java stream, and

wherein the third Java stream provides for the transmission of data according to a specific communication protocol layer.

9. – 12. (CANCELED)

13. (CURRENTLY AMENDED) A computer data signal embodied in a carrier wave and representing sequences of instructions embodied on a non-transitory computer-readable medium which, when executed by one or more processors, provide communication protocol layer independent security for data transmitted between a first process, executing on a first network node, and a second process, executing on a second network node, according to at least one common communication protocol layer supported by the first and second network nodes, by performing the steps of:

establishing a Java secure ~~communication~~ channel between the first network node and the second network node;

establishing a ~~fist~~ first Java stream between the first process and the Java secure ~~communication~~ channel;

establishing a second Java stream between the second process and the Java secure ~~communication~~ channel;

in response to the data being written to the first Java stream, encrypting the data to generate encrypted data, the encrypting of the data being performed independent of any communication protocol layers used to transport the encrypted data from the first network node to the second network node;

causing the encrypted data to be transmitted from the first network node to the second network node according to the at least one communication protocol layer supported by the first and second network nodes; and

in response to the encrypted data being read from the second Java stream, decrypting the encrypted data to recover decrypted data which is identical to the data on the first network node before the data was written to the first stream, the decrypting of the encrypted data being performed independent of any communication protocol layers used to transport the encrypted data from the first network node to the second network node.

14. (PREVIOUSLY PRESENTED) The computer data signal of Claim 13, wherein the computer sequence of instructions further includes instructions for performing the steps of

performing a communication protocol layer specific encryption of the data on the first network node, and

performing a communication protocol layer specific decryption of the data on the second network node.

15. (CANCELED)

16. (CURRENTLY AMENDED) The computer data signal of Claim 13, ~~wherein the communication channel is a Java secure channel,~~

~~wherein the first stream is a Java stream,~~

~~wherein the second stream is a Java stream,~~

wherein the computer sequence of instructions further includes instructions for connecting the Java secure channel to a third Java stream, and

wherein the third Java stream provides for the transmission of data according to a specific communication protocol layer.

17. (CURRENTLY AMENDED) A method for providing communication protocol layer independent security for data transmitted by a process executing on a network node, the method comprising the steps of:

a) establishing a first Java stream between the process and a Java secure ~~communication~~ channel; and

b) in response to the data being written to the first Java stream, encrypting the data to generate encrypted data, the encrypting of the data being performed independent of any communication protocol layers used to transport the encrypted data on the Java secure ~~communication~~ channel.

18. (CANCELED)

19. (CURRENTLY AMENDED) The method of claim 17, ~~wherein the communication channel is a Java secure channel, wherein the stream is a Java stream,~~

wherein the method further comprises the step of connecting the Java secure channel to a second Java stream, and

wherein the second Java stream provides for the transmission of data according to a specific communication protocol layer.

20. (CURRENTLY AMENDED) A method for providing communication protocol-independent security for data transmitted between a first node and a second node, the method comprising the steps of:

establishing a Java secure ~~communication~~ channel between a first network node and a second network node;

establishing a first Java stream from a first process to the Java secure ~~communication~~ channel after the establishment of the Java secure ~~communication~~ channel, wherein the first Java stream is encrypted after the first process and before entering the Java secure ~~communication~~ channel and the encrypted first Java stream is independent of any communication protocol layers; and

establishing a second Java stream from the Java secure ~~communication~~ channel to a second process after the establishment of the Java secure ~~communication~~ channel, wherein the second Java stream is decrypted after the Java secure ~~communication~~ channel and before entering the second process.

21. (CANCELED)

22. (CANCELED)

23. (CURRENTLY AMENDED) The method of clam 20, further comprising ~~wherein;~~

~~the communication channel is a Java secure channel;~~

~~the first stream is a Java stream;~~

~~the second stream is a Java stream;~~

~~the method further comprises the step of~~ connecting the Java secure channel to a third Java stream~~;~~, wherein ~~and~~

the third Java stream provides for the transmission of data according to a specific communication protocol layer.

24. (CURRENTLY AMENDED) A non-transitory computer-readable medium carrying one or more sequences of one or more instructions for providing communication protocol-layer independent security for data transmitted between a first node and a second node, the one or more sequences of one or more instructions including instructions which, when executed by one or more processors, cause the one or more processors to perform the steps of:

establishing a Java secure ~~communication~~ channel between a first network node and a second network node;

establishing a first Java stream from a first process to the Java secure ~~communication~~ channel after the establishment of the Java secure ~~communication~~ channel, wherein the first Java stream is encrypted after the first process and before entering the Java secure ~~communication~~ channel and the encrypted first Java stream is independent of any communication protocol layers; and

establishing a second Java stream from the Java secure ~~communication~~ channel to a second process after the establishing of theJava secure ~~communication~~ channel, wherein the second Java stream is decrypted after the Java secure ~~communication~~ channel and before entering the second process.

25. (CANCELED)

26. (CANCELED)

27. (CURRENTLY AMENDED) The non-transitory computer-readable medium ~~method~~ of claim 24, wherein the non-transitory computer-readable medium further includes instructions for :

~~the communication channel is a Java secure channel;~~

~~the first stream is a Java stream;~~

~~the second stream is a Java stream~~

~~the method further comprises the step of~~ connecting the Java secure channel to a third Java stream~~;~~, wherein ~~and~~

the third Java stream provides for the transmission of data according to a specific communication protocol layer.

28. (CURRENTLY AMENDED) A communications network providing communication protocol-independent security for data ~~transmitted between the first node and a second node~~, comprising~~the communication network performing the steps of~~:

a first network node;

a second network node;

~~establishing~~ a ~~communication~~ Java secure channel between the ~~a~~ first network node and the ~~a~~ second network node;

~~establishing~~ a first Java stream between ~~from~~ a first process of the first node and ~~to~~ the Java secure ~~communication~~ channel ~~after the establishment of the communication channel~~, wherein ~~the first stream is encrypted after the first process and before entering the communication channel and~~ the first process encrypts the ~~encrypted~~ first Java stream ~~is~~ independent of any communication protocol layers; and

~~establishing~~ a second Java stream between ~~from~~ the Java secure ~~communication~~ channel and ~~to~~ a second process of the second node~~after the establishment of the communication channel~~, wherein the second process decrypts the second Java stream ~~is decrypted after the communication channel and before entering the second process~~.

29. (CURRENTLY AMENDED) The communication network of claim 28, wherein at least one of the encryption of the first stream and the decryption of the second stream is specific to a communication protocol layer.

30. (CANCELED)

31. (CURRENTLY AMENDED) The communication network of claim 28, further comprising

~~wherein:~~

~~the communication channel is a Java secure channel;~~

~~the first stream is a Java stream;~~

~~the second stream is a Java stream~~

~~the method further comprises the step of connecting the Java secure channel to~~ a third Java stream that ~~; and~~

~~the third Java stream~~ provides for the transmission of data according to a specific communication protocol layer.

32. (CURRENTLY AMENDED) A computer data signal embodied in a carrier wave and representing sequences of instructions embodied in a non-transitory computer-readable medium which, when executed by one or more processor, provide communication protocol-independent security for data transmitted between a first node and second node, by performing the steps of:

establishing a Java secure ~~communication~~ channel between a first network node and a second network node;

establishing a first Java stream from a first process to the Java secure ~~communication~~ channel after the establishment of the Java secure ~~communication~~ channel, wherein the first Java stream is encrypted after the first process and before entering the Java secure ~~communication~~ channel and the encrypted first Java stream is independent of any communication protocol layers; and

establishing a second Java stream from the Java secure ~~communication~~ channel to a second process after the establishment of the Java secure ~~communication~~ channel, wherein the second Java stream is decrypted after the Java secure ~~communication~~ channel and before entering the second process.

33. (CANCELED)

34. (CANCELED)

35. (CURRENTLY AMENDED) The computer data signal of claim 32, wherein the computer sequence of instructions further includes instructions for :

~~the communication channel is a Java secure channel;~~

~~the first stream is a Java stream;~~

~~the second stream is a Java stream~~

~~the method further comprises the step of~~ connecting the Java secure channel to a third Java stream~~;~~, wherein ~~and~~

the third Java stream provides for the transmission of data according to a specific communication protocol layer.